

**BCSE103E -**  
**Computer Programming: Java Lab**  
**LAB Assessment – 2**

**Name:** Dhruv Rajeshkumar Shah  
**Registration No :** 21BCE0611  
**Date :** 2<sup>nd</sup> November 2022

## Question 1

Write a class named '**Student**' with the following:

- Data members include **StudentID**, **FName** and **LName** and **Age**.
- Constructor with argument to initialize the data members.
- A method named '**getDetails**' to read the details of the student.

Write a class Named **MarksAnalysis** that inherits **Student** with the following

- Data members named **Total**, **Average**.
- A method '**findGrade**' will accept the 5 set of marks from the user and find the **Total**, **Average** and **grade** based on the following table; And print total, average and all the marks and their corresponding grades

| Total    | Grade |
|----------|-------|
| 90 – 100 | S     |
| 80 – 89  | A     |
| 70 – 79  | B     |
| 60 – 69  | C     |
| 50 – 59  | D     |
| <50      | E     |

- A method named '**printDetails**' to display StudentID, FName and LName and Age.
- Start the program execution in the **MarksAnalysis** class and display the student details using the **printDetails** method.

# SCREENSHOTS

VIT LMS

BCSE103E\_VL2022230104641

Participants

Competencies

Grades

General

Assessment 1

**Assessment 2**

Quiz

Debugging

File Submission

FAT

Dashboard

Site home

Calendar

BCSE103E Computer Programming: Java (Lab) Fall 2022-23  
(L37+L38+L45+L46) [VL2022230104641]

Dashboard / My courses / SCOPE / BCSE103E\_VL2022230104641 / Assessment 2 / que 1

Description Submission <> Edit Submission view

Submitted on Wednesday, 2 November 2022, 5:04 PM (Download) (Evaluate)

MarksAnalysis.java

```
1 // Assessment 2
2 // Dhruv Rajeshkumar Shah
3 // 21BCE0611
4
5 import java.util.Scanner;
6
7 // Student class
8 class Student {
9     int StudentID, Age;
10    String FName, LName;
11
12    // Constructors
13    Student() {
14        this.StudentID = 0;
15        this.FName = "";
16        this.LName = "";
17        this.Age = 0;
18    }
19
20    Student(int StudentID, String FName, String LName, int Age) {
21        this.StudentID = StudentID;
22        this.FName = FName;
23        this.LName = LName;
24        this.Age = Age;
25    }
26
27    void printDetails() {
28        System.out.println("Student ID: " + this.StudentID);
29        System.out.println("First Name: " + this.FName);
30        System.out.println("Last Name: " + this.LName);
31        System.out.println("Age: " + this.Age);
32    }
33
34    void finalGrade() {
35        Scanner sc = new Scanner(System.in);
36        int sub1 = sc.nextInt();
37        int sub2 = sc.nextInt();
38        int sub3 = sc.nextInt();
39        int sub4 = sc.nextInt();
40        int sub5 = sc.nextInt();
41
42        this.Total = sub1 + sub2 + sub3 + sub4 + sub5;
43        this.Average = this.Total / 5;
44
45        String grade;
46        if (this.Average >= 90) {
47            grade = "S";
48        } else if (this.Average >= 80) {
49            grade = "A";
50        } else if (this.Average >= 70) {
51            grade = "B";
52        } else if (this.Average >= 60) {
53            grade = "C";
54        } else if (this.Average >= 50) {
55            grade = "D";
56        } else {
57            grade = "E";
58        }
59
60        System.out.println("Total Marks: " + this.Total);
61        System.out.println("Average Marks: " + this.Average);
62        System.out.println("Grade: " + grade);
63    }
64
65    // Print Details
66    void printDetails() {
67        System.out.println("Student ID: " + this.StudentID);
68        System.out.println("First Name: " + this.FName);
69        System.out.println("Last Name: " + this.LName);
70        System.out.println("Age: " + this.Age);
71    }
72
73    public static void main(String[] args) {
74        MarksAnalysis obj = new MarksAnalysis();
75        obj.getDetails();
76        obj.printDetails();
77        obj.finalGrade();
78    }
79
80 }
```

MarksAnalysis.java

```
67 sub1 = sc.nextInt();
68
69 this.Total = sub1 + sub2 + sub3 + sub4 + sub5;
70 this.Average = this.Total / 5;
71
72 String grade;
73 if (this.Average >= 90) {
74     grade = "S";
75 } else if (this.Average >= 80) {
76     grade = "A";
77 } else if (this.Average >= 70) {
78     grade = "B";
79 } else if (this.Average >= 60) {
80     grade = "C";
81 } else if (this.Average >= 50) {
82     grade = "D";
83 } else {
84     grade = "E";
85 }
86
87 System.out.println("Total Marks: " + this.Total);
88 System.out.println("Average Marks: " + this.Average);
89 System.out.println("Grade: " + grade);
90
91 }
92
93 // Print Details
94 void printDetails() {
95     System.out.println("Student ID: " + this.StudentID);
96     System.out.println("First Name: " + this.FName);
97     System.out.println("Last Name: " + this.LName);
98     System.out.println("Age: " + this.Age);
99 }
100
101 public static void main(String[] args) {
102     MarksAnalysis obj = new MarksAnalysis();
103     obj.getDetails();
104     obj.printDetails();
105     obj.finalGrade();
106 }
107
108 }
109 }
```

Console: connection closed (Running: 16 seg)

```
Enter Student ID: 0611
Enter First Name: Dhruv
Enter Last Name: Shah
Enter Age: 19
Student ID: 611
First Name: Dhruv
Last Name: Shah
Age: 19
Enter Marks in Subject 1: 90
Enter Marks in Subject 2: 96
Enter Marks in Subject 3: 99
Enter Marks in Subject 4: 92
Enter Marks in Subject 5: 80
Total Marks: 457
Average Marks: 91.0
Grade: S
```

Description

Write a class named **'Student'** with the following:

- Data members include **StudentID, FName** and **LName** and **Age**.
- Constructor with argument to initialize the data members.
- A method named **'getDetails'** to read the details of the student.

Write a class Named **MarksAnalysis** that inherits **Student** with the following

- Data members named **Total, Average**.
- A method **'findGrade'** will accept the 5 set of marks from the user and find the **Total, Average** and **grade** based on

People are waiting in the lobby

Return to the main meeting window to see who's waiting.

## CODE

```
// Assessment 2
// Dhruv Rajeshkumar Shah
// 21BCE0611

import java.util.Scanner;

// Student class
class Student {
    int StudentID, Age;
    String Fname, Lname;

    // Constructors
    Student() {
        this.StudentID = 0;
        this.Fname = "";
        this.Lname = "";
        this.Age = 0;
    }

    Student(int StudentID, String Fname, String Lname, int Age) {
        this.StudentID = StudentID;
        this.Fname = Fname;
        this.Lname = Lname;
        this.Age = Age;
    }

    // Get Details
    void getDetails() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter Student ID: ");
        this.StudentID = sc.nextInt();
        System.out.print("Enter First Name: ");
        this.Fname = sc.next();
        System.out.print("Enter Last Name: ");
        this.Lname = sc.next();
        System.out.print("Enter Age: ");
        this.Age = sc.nextInt();
        // sc.close();
    }
}

public class MarksAnalysis extends Student {
    int Total;
    float Average;

    // Constructor
```

```

MarksAnalysis() {
    super();
    this.Total = 0;
    this.Average = 0;
}

// Final Grade
void finalGrade() {
    int sub1, sub2, sub3, sub4, sub5;
    Scanner sc1 = new Scanner(System.in);
    System.out.print("Enter Marks in Subject 1: ");
    sub1 = sc1.nextInt();
    System.out.print("Enter Marks in Subject 2: ");
    sub2 = sc1.nextInt();
    System.out.print("Enter Marks in Subject 3: ");
    sub3 = sc1.nextInt();
    System.out.print("Enter Marks in Subject 4: ");
    sub4 = sc1.nextInt();
    System.out.print("Enter Marks in Subject 5: ");
    sub5 = sc1.nextInt();

    this.Total = sub1 + sub2 + sub3 + sub4 + sub5;
    this.Average = this.Total / 5;

    String grade;
    if (this.Average >= 90) {
        grade = "S";
    } else if (this.Average >= 80) {
        grade = "A";
    } else if (this.Average >= 70) {
        grade = "B";
    } else if (this.Average >= 60) {
        grade = "C";
    } else if (this.Average >= 50) {
        grade = "D";
    } else {
        grade = "E";
    }

    System.out.println("Total Marks: " + this.Total);
    System.out.println("Average Marks: " + this.Average);
    System.out.println("Grade: " + grade);

}

// Print Details
void printDetails() {
    System.out.println("Student ID: " + this.StudentID);
}

```

```

        System.out.println("First Name: " + this.Fname);
        System.out.println("Last Name: " + this.Lname);
        System.out.println("Age: " + this.Age);
    }

    // Main
    public static void main(String[] args) {
        MarksAnalysis obj = new MarksAnalysis();
        obj.getDetails();
        obj.printDetails();
        obj.finalGrade();
    }
}

```

## Question 7

Prime Exception

Create Four user defined exceptions 1. PrimeException 2. NotPrimeException  
3.OddException 4.EvenException

Write a program to read numbers from the user and find whether it is prime or not prime and odd or even and throw corresponding exception and handle them with appropriate message to the user.

Sample

input: 10

output: not prime exception thrown

even exception thrown

# SCREENSHOTS

The screenshot shows the VIT LMS interface. On the left is a sidebar with navigation options: BCSE103E\_VL2022230104641, Participants, Competencies, Grades, General, Assessment 1, **Assessment 2**, Quiz, Debugging, File Submission, FAT, Dashboard, Site home, and Calendar. The main content area displays the course title "BCSE103E Computer Programming: Java (Lab) Fall 2022-23 (L37+L38+L45+L46) [VL2022230104641]" and a breadcrumb trail: Dashboard / My courses / SCOPE / BCSE103E\_VL2022230104641 / Assessment 2 / que 7. Below this, there are tabs for Description, Submission, Edit, and Submission view. The Submission tab is active, showing a submission status of "Submitted on Wednesday, 2 November 2022, 5:01 PM" with options to Download and Evaluate. An "Automatic evaluation" section shows the code for "ExceptionHandling.java".

```
1 // Assessment 2
2 // Dhruv Rajeshkumar Shah
3 // 21BCE0611
4
5 import java.util.Scanner;
6
7 // User defined exceptions
8
9 // Prime exception
10 class PrimeException extends Exception {
11     @Override
12     public String toString() {
13         return "prime exception thrown";
14     }
15 }
16
17 // Not prime exception
18 class NotPrimeException extends Exception {
19     @Override
20     public String toString() {
```

The screenshot shows an IDE with a Java code file named "ExceptionHandling.java". The code includes a flag variable, a checkOddOrEven method, and a main method. A console window is open, showing the output of the program. The output indicates that the user entered the number 10, and the program threw a "not prime exception" and an "even exception".

```
47 // flag = true;
48 // break;
49 }
50
51 if (flag) {
52     throw new NotPrimeException();
53 } else {
54     throw new PrimeException();
55 }
56 }
57
58 // Check odd or even
59 static void checkOddOrEven(int n) throws OddException, EvenException {
60     if (n % 2 == 0) {
61         throw new EvenException();
62     } else {
63         throw new OddException();
64     }
65 }
66
67 // Main
68 public static void main(String[] args) {
69     Scanner sc = new Scanner(System.in);
70     System.out.print("Enter a number: ");
71     int n = sc.nextInt();
72     try {
73         checkPrime(n);
74     } catch (PrimeException e) {
75         System.out.println(e);
76     } catch (NotPrimeException e) {
77         System.out.println(e);
78     }
79     try {
80         checkOddOrEven(n);
81     } catch (OddException e) {
82         System.out.println(e);
83     } catch (EvenException e) {
84         System.out.println(e);
85     }
86     sc.close();
87 }
88 }
89 }
```

Console: connection closed (Running: 1 seg)

```
Enter a number: 10
not prime exception thrown
even exception thrown
```

## CODE

```
// Assessment 2
// Dhruv Rajeshkumar Shah
// 21BCE0611

import java.util.Scanner;

// User defined exceptions

// Prime exception
class PrimeException extends Exception {
    @Override
    public String toString() {
        return "prime exception thrown";
    }
}

// Not prime exception
class NotPrimeException extends Exception {
    @Override
    public String toString() {
        return "not prime exception thrown";
    }
}

// Odd exception
class OddException extends Exception {
    @Override
    public String toString() {
        return "odd exception thrown";
    }
}

// Even exception
class EvenException extends Exception {
    @Override
    public String toString() {
        return "even exception thrown";
    }
}

public class ExceptionHandling {
    // Check prime or not prime
    static void checkPrime(int n) throws PrimeException, NotPrimeException {
        boolean flag = false;
        for (int i = 2; i <= n / 2; ++i) {
            if (n % i == 0) {
                flag = true;
            }
        }
    }
}
```



```

        break;
    }
}
if (flag) {
    throw new NotPrimeException();
} else {
    throw new PrimeException();
}
}

// Check odd or even
static void checkOddOrEven(int n) throws OddException, EvenException {
    if (n % 2 == 0) {
        throw new EvenException();
    } else {
        throw new OddException();
    }
}

// Main
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter a number: ");
    int n = sc.nextInt();
    try {
        checkPrime(n);
    } catch (PrimeException e) {
        System.out.println(e);
    } catch (NotPrimeException e) {
        System.out.println(e);
    }
    try {
        checkOddOrEven(n);
    } catch (OddException e) {
        System.out.println(e);
    } catch (EvenException e) {
        System.out.println(e);
    }
    sc.close();
}
}

```